EE2703 : Applied Programming Lab
Endsem
Finding Antenna Currents in a Half Wave Dipole Antenna

Sai Shashank GP
EE20B040

May 13, 2022

Introduction

We are asked to validate the model of current in Half Wave Dipole Antenna which is given as follows

$$I = \left\{ \begin{array}{ll} I_m \sin(k(l+z)) & -l \leq z \leq 0 \\ I_m \sin(k(l-z)) & 0 \leq z \leq l \end{array} \right\}$$

So, we will be dividing the antenna into parts, find currents in each part and plot the model currents to compare the validity of model. We will be using pylab for plotting, numpy for matrix and vector operations.

Assignment

Prerequisites

We have to first import all the libraries and define all the constants we need and use.

```
import numpy as np
from pylab import *

# define all the constants we need

l = 0.5
c = 2.9979e8
mu0 = 4*np.pi*1e-7
N = 4
# N = 100
Im = 1.0
a = 0.01
lambda_ = l*4
f = c/lambda_
k = 2*np.pi/lambda_
dz = l/N
```

Q1: Setting the currents and coordinates

We have to initialise the z-coordinates of the known and unknown currents. After that, we have to also calculate the actual current to plot later. It is done as follows.

```
i = np.arange(-N, N+1, 1)
z = i*dz
j1 = np.arange(-N+1, 0, 1)
j2 = np.arange(1, N, 1)
j = np.r_[j1, j2]
u = j*dz
I = np.zeros(2*N +1)
I[N] = Im
J = np.zeros(2*N -2)
I_actual = Im*np.sin(k*(l+np.abs(z)))
```

Q2: Calculate M matrix

Calculating the M matrix is from Ampere's law. It is from the following equation.

$$2\pi a H_\phi(z_i) = I_i$$

Writing it in matrix form, we get, M matrix as follows $\frac{1}{2\pi a}$I

```
def computeM(a, N):
    const = 1/(2*np.pi*a)
    M = const*np.identity(2*N -2, dtype=float)
    return M

M = computeM(a, N)
```

Final Equation for magnetic field is as follows in matrices

$$H_{phi} = M * J$$

Q3: Calculate Magnetic Vector Potential

Calculating Magnetic Vector Potential will help us find Magnetic Field in another way. It is done as follows

$$A_{z,i} = \sum_j P_{ij} I_j + P_B I_N$$

$$P_{ij} = \frac{\mu_0}{4\pi} \frac{exp(-jkR_{ij})}{R_{ij}} dz'_j$$

$$P_B = \frac{\mu_0}{4\pi} \frac{exp(-jkR_{iN})}{R_{iN}} dz'_j$$

and R_{ij} is the distance from source at (0, z'_i) and observer at (r_j, z_j). To find R matrix, I have implemented the following pseudo code

```
zz, zz' = meshgrid(z, z')
rr = meshgrid(r,r)[0]
R = rr + 1j*(zz-zz')
return abs(R)
```

After finding R, we can find P and P_B can be found out easily.

```
def calcP(R):
    const = mu0/(4*np.pi)
    P = const*np.exp(-1j*k*R)*dz/R
    return P
def calcPB(Rn):
    const = mu0/(4*np.pi)
    Rn = list(Rn)
    Rn.pop(N)
    R = np.array(Rn[1:-1])
    P = -1*const*np.exp(-1j*k*R)*dz/R
    return P
```

Q4: Calculating Magnetic Field from above result

Magnetic field can be calculated as follows

$$H_\phi(r, z_i) = \sum_j Q_{ij} I_j + Q_B I_N$$

where

$$Q_{ij} = -P_{ij} \frac{r}{\mu_0} \left(\frac{-jk}{R_{ij}} - \frac{1}{{R_{ij}}^2} \right)$$

$$Q_B = -P_B \frac{r}{\mu_0} \left(\frac{-jk}{R_{iN}} - \frac{1}{{R_{iN}}^2} \right)$$

So, calculating them is as follows

```
def calcQ(P, R, r):
    const1 = 1/mu0
    const2 = (-1/R**2) + 1j*(-k/R)
    return -1*P*r*const1*const2

def calcQB(PB, Rn, r):
    const1 = 1/mu0
    Rn = list(Rn)
    Rn.pop(N)
    R = np.array(Rn[1:-1])
    const2 = (-1/R**2) + 1j*(-k/R)
    return -1*PB*r*const1*const2
```

Q5: Calculating the currents and printing the outputs

Now, calculating currents is easy. It can be done from the following equation of matrices

$$MJ = QJ + Q_B I_m$$

So, J is as follows

$$J = (M - Q)^{-1}.Q_B I_m$$

The code is as follows,

```
A = inv(M-Q)
B = QB*Im
J = np.matmul(A, B)
```

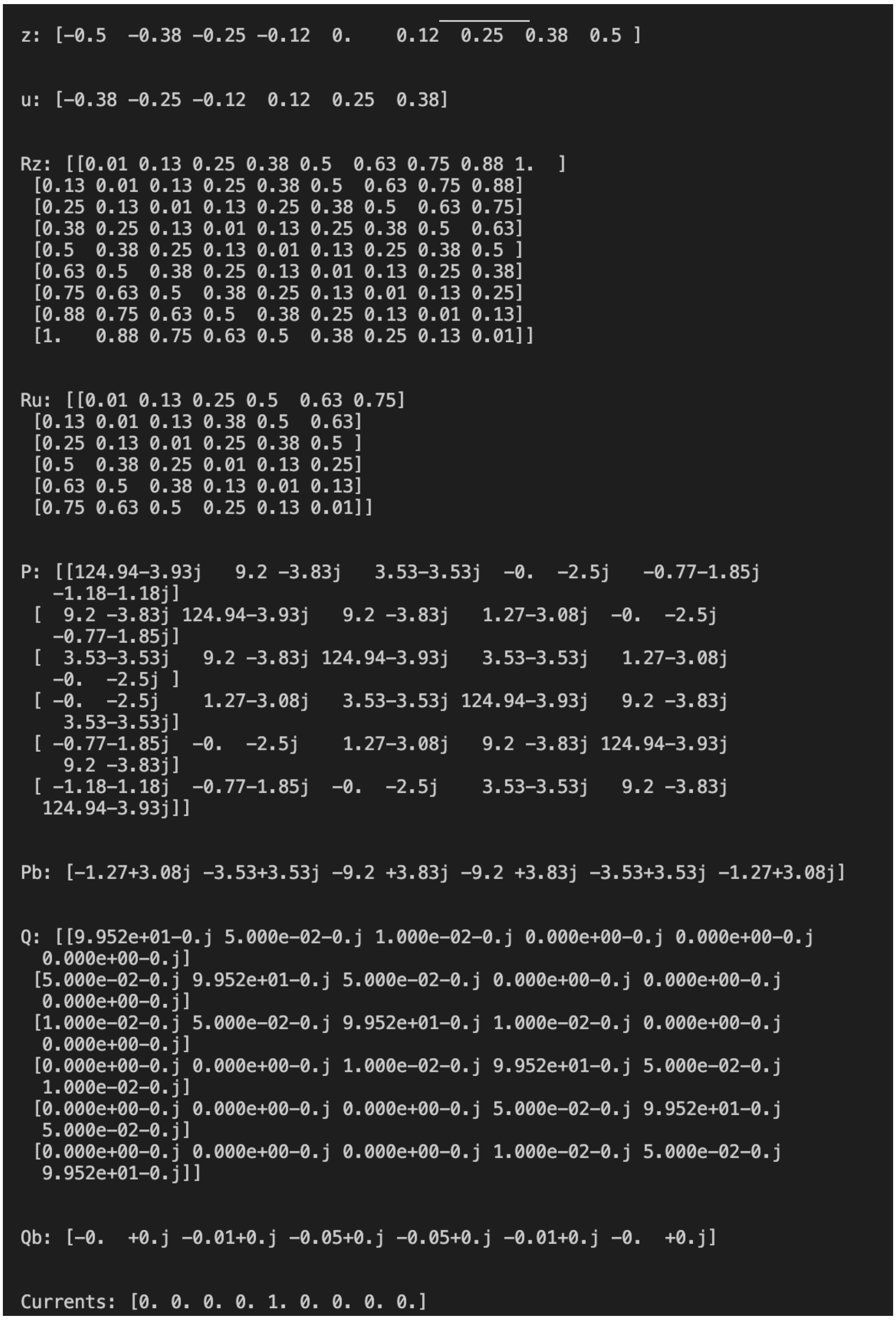
Printing the outputs gave me this in my terminal

Plotting the currents

We have to insert the known values of currents into the now-known currents array. That can be done using this code

```
J = list(J)
J.insert(0, 0)
J.insert(2*N-1, 0)
J.insert(N, Im)
```

After this we can plot the currents calculated with the actual current values



```
z: [-0.5  -0.38 -0.25 -0.12  0.    0.12  0.25  0.38  0.5 ]


u: [-0.38 -0.25 -0.12  0.12  0.25  0.38]


Rz: [[0.01 0.13 0.25 0.38 0.5  0.63 0.75 0.88 1.  ]
 [0.13 0.01 0.13 0.25 0.38 0.5  0.63 0.75 0.88]
 [0.25 0.13 0.01 0.13 0.25 0.38 0.5  0.63 0.75]
 [0.38 0.25 0.13 0.01 0.13 0.25 0.38 0.5  0.63]
 [0.5  0.38 0.25 0.13 0.01 0.13 0.25 0.38 0.5 ]
 [0.63 0.5  0.38 0.25 0.13 0.01 0.13 0.25 0.38]
 [0.75 0.63 0.5  0.38 0.25 0.13 0.01 0.13 0.25]
 [0.88 0.75 0.63 0.5  0.38 0.25 0.13 0.01 0.13]
 [1.   0.88 0.75 0.63 0.5  0.38 0.25 0.13 0.01]]


Ru: [[0.01 0.13 0.25 0.5  0.63 0.75]
 [0.13 0.01 0.13 0.38 0.5  0.63]
 [0.25 0.13 0.01 0.25 0.38 0.5 ]
 [0.5  0.38 0.25 0.01 0.13 0.25]
 [0.63 0.5  0.38 0.13 0.01 0.13]
 [0.75 0.63 0.5  0.25 0.13 0.01]]


P: [[124.94-3.93j   9.2 -3.83j   3.53-3.53j  -0.  -2.5j   -0.77-1.85j
   -1.18-1.18j]
 [  9.2 -3.83j 124.94-3.93j   9.2 -3.83j   1.27-3.08j  -0.  -2.5j
   -0.77-1.85j]
 [  3.53-3.53j   9.2 -3.83j 124.94-3.93j   3.53-3.53j   1.27-3.08j
   -0.  -2.5j ]
 [ -0.  -2.5j    1.27-3.08j   3.53-3.53j 124.94-3.93j   9.2 -3.83j
    3.53-3.53j]
 [ -0.77-1.85j  -0.  -2.5j    1.27-3.08j   9.2 -3.83j 124.94-3.93j
    9.2 -3.83j]
 [ -1.18-1.18j  -0.77-1.85j  -0.  -2.5j    3.53-3.53j   9.2 -3.83j
  124.94-3.93j]]


Pb: [-1.27+3.08j -3.53+3.53j -9.2 +3.83j -9.2 +3.83j -3.53+3.53j -1.27+3.08j]


Q: [[9.952e+01-0.j 5.000e-02-0.j 1.000e-02-0.j 0.000e+00-0.j 0.000e+00-0.j
  0.000e+00-0.j]
 [5.000e-02-0.j 9.952e+01-0.j 5.000e-02-0.j 0.000e+00-0.j 0.000e+00-0.j
  0.000e+00-0.j]
 [1.000e-02-0.j 5.000e-02-0.j 9.952e+01-0.j 1.000e-02-0.j 0.000e+00-0.j
  0.000e+00-0.j]
 [0.000e+00-0.j 0.000e+00-0.j 1.000e-02-0.j 9.952e+01-0.j 5.000e-02-0.j
  1.000e-02-0.j]
 [0.000e+00-0.j 0.000e+00-0.j 0.000e+00-0.j 5.000e-02-0.j 9.952e+01-0.j
  5.000e-02-0.j]
 [0.000e+00-0.j 0.000e+00-0.j 0.000e+00-0.j 1.000e-02-0.j 5.000e-02-0.j
  9.952e+01-0.j]]


Qb: [-0.  +0.j -0.01+0.j -0.05+0.j -0.05+0.j -0.01+0.j -0.  +0.j]


Currents: [0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

Figure 1: Intermediate and Current arrays for N = 4

Observations and Conclusions

I have observed that for increase in N, the curve for calculated currents are following this pattern.

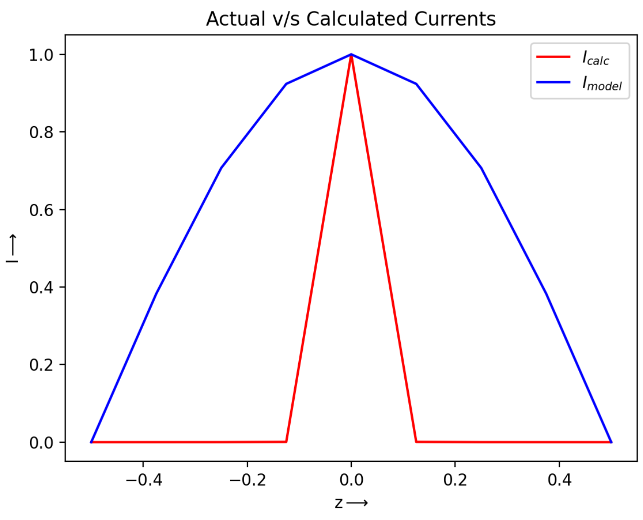


Figure 2: Actual v/s Calculated currents for N = 4

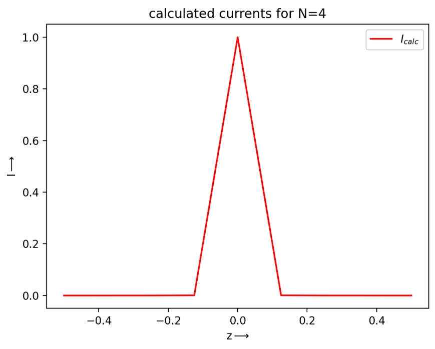


Figure 3: calculated currents N=4

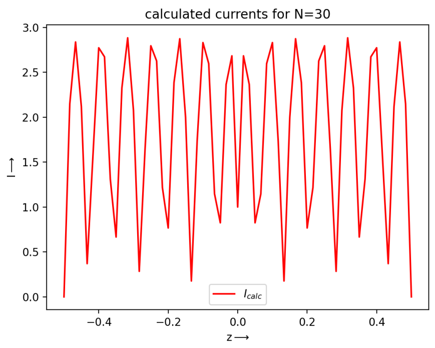


Figure 4: calculated currents N=30

The optimum fit occurs at N = 100-150. I'm thinking that this behaviour of function variation with sampling is bacuase of sampling affecting the phase of complex exponential in P and P_B
In conclusion, I think this model is a good fit for the Antenna currents in a Half Wave Dipole Antenna. There is scope for betterment.

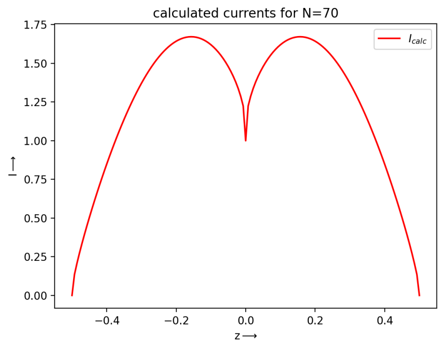


Figure 5: calculated currents N=70

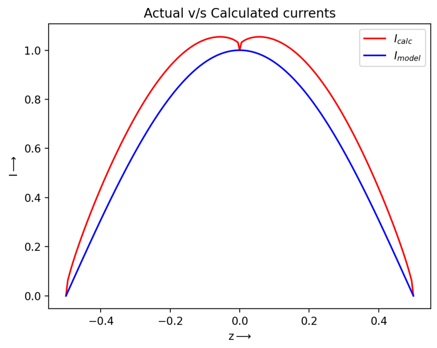


Figure 6: calculated currents N=100

TakeAways

By doing this assignment, I have learnt the following

- How can matrices and vector operations can reduce the use of for loops
- How can a function which involves complex exponentials behave with sampling.
- Some basic concepts of Dipole Antenna